Cooperative Perception Using V2X Communications: An Experimental Study

Faisal Hawlader, François Robinet and Raphaël Frank
Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg, 29 Avenue J.F Kennedy, L-1855 Luxembourg
firstname.lastname@uni.lu

2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall) | 979-8-3315-1778-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/VTC2024-Fall63153.2024.10757448

***Abstract*—The development and testing of real-time perception capabilities is crucial to fully realize the potential of connected and automated driving. This study presents a comprehensive analysis of end-to-end delay and object detection quality for distributed perception. We investigate object detection from camera images in two distinct scenarios: local and cloud processing. In the local scenario, objects are detected using only on-board hardware. The results are then used to generate Cooperative Perception Messages (CPM), which are broadcast to nearby vehicles via the ITS-G5 communication technology. In the cloud processing scenario, images are compressed using H.265 before being transmitted to the cloud via C-V2X, where objects are detected on more demanding hardware. The detection results are then transmitted to vehicles in the vicinity, enabling cooperative perception scenarios. This study evaluates real-time object detection systems in real-world conditions, and highlights the trade-offs between the end-to-end detection delay and quality. By leveraging emerging technologies such as ITS-G5 and C-V2X, our research provides crucial insights for the development of efficient perception systems in the connected driving ecosystem.**

***Index Terms*—Cloud Computing; Distributed Perception; C-V2X; ITS-G5; Object Detection; CPM**

I. Introduction

Real-time perception is critical for ensuring the safety and efficiency of autonomous driving systems [1, 2]. However, the computational demands of perception systems often exceed the capabilities of vehicles with limited computing resources, leading to a trade-off between accuracy and end-to-end delay [3, 4]. To address this challenge, offloading perception tasks to cloud servers has emerged as a promising solution [5]. By shifting computationally intensive processing to the cloud and conducting lighter computations onboard the vehicle, resources can be allocated more efficiently [6]. However, offloading introduces latency, particularly in the transmission of raw sensor data between the vehicle and the cloud, posing significant challenges for real-time object detection. To mitigate these challenges, existing research commonly employs various data compression strategies [7, 8]. These strategies aim to reduce network bandwidth usage, transmission latency, and computing power consumption on board. However, while these approaches offer potential benefits, they also raise concerns about potential accuracy loss [9]. Achieving the optimal balance between end-to-end delay, accuracy, and computational efficiency requires careful investigation [2, 6, 10, 11].

Our study sheds light on the trade-offs between end-to-end delay and detection quality, offering insights into the real-world performance of real-time object detection systems. In particular, we make the following contributions:

1) We establish a validation framework tailored to assess the performance of real-time perception in connected driving environments. This framework includes field trials and validation methodologies specifically designed to address the challenges of processing sensor data.
2) We create and release a real-world driving dataset using a front-facing rooftop camera, annotating pedestrian, vehicle, and traffic light classes [1]. Using this dataset, we train object detectors to establish a perception pipeline used to evaluate cooperative driving scenarios.
3) We explore both local and cloud processing techniques to optimize real-time object detection performance, minimize latency, and ensure robustness in real-world scenarios. To offload processing to the cloud, we benchmark the use of H.265 video encoding and study the trade-off between end-to-end delay and detection quality.

The remainder of this paper is structured as follows. In Section II, we review the related literature. Section III describes experimental setup, including details on the hardware and software configuration, the vehicle instrumentation, and data collection during field trials. Section IV presents the experimental results and discusses data processing trade-offs in terms of end-to-end delay and detection quality. Finally, in Section V, we conclude this work with a summary of our findings.

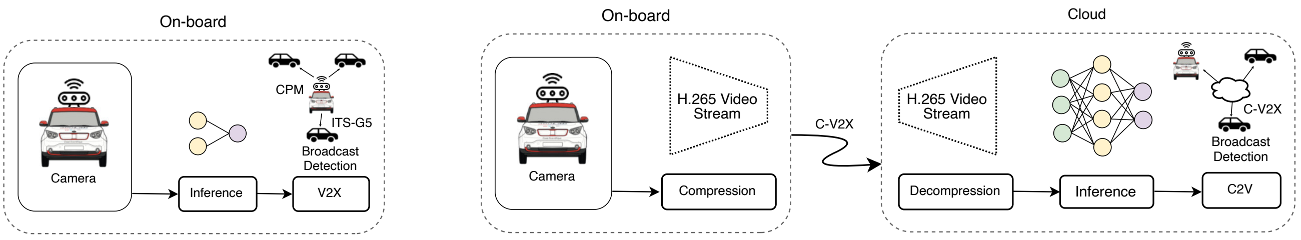
II. Related Work

In this section, we review the existing literature relevant to real-time perception in connected and automated driving, focusing on key areas related to our study: Previous studies [12, 13] have highlighted the importance of Vehicle-to-Everything (V2X) communication in improving the range of perception [14] and fostering the evolution of connected driving ecosystems [6]. Recent works have explored various dimensions of V2X and Cooperative Perception Messages (CPM), including communication protocols [15], message formats [12, 15]. Protocols such as ITS-G5 [16] and C-V2X [17] have been extensively investigated for their efficacy in enabling low latency and reliable communication channels between vehicles [4, 18]. In addition, studies have

[1] **Dataset available:** https://github.com/FaisalHawlader/LuxDrive-Dataset

979-8-3315-1778-6/24/$31.00 ©2024 IEEE

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.



(a) Local scenario

(b) Cloud scenario

Fig. 1: Experimental scenarios: in the Local scenario, inference is performed on-board using a lightweight detection model, and the results are transmitted as CPMs to nearby vehicles and/or infrastructures over ITS-G5. In the Cloud scenario, a compressed video feed is sent to a Cloud server using C-V2X, where a more computationally demanding detector is used. The detection results are then transmitted to vehicles in the vicinity, enabling a cooperative perception scenario.

focused particularly on designing CPM formats and establishing message generation rules [19] to optimize the exchange of relevant perception data while minimizing communication overhead [12, 20]. However, on-board perception systems in vehicles equipped with constrained computing resources encounter challenges due to high computational demands, as demonstrated in previous studies [5, 19]. Achieving real-time performance is impeded by the necessity to strike a balance between accuracy and processing speed, as highlighted in recent surveys [2]. As shown in prior works [21, 22], the integration of cloud technology with C-V2X shows considerable potential by enabling vehicles to transmit sensor data to a remote cloud server for processing, thereby leveraging the abundant resources available therein [23]. Research has shown [24] that using C-V2X communication for cloud-based processing can further improve scalability and computational capabilities, facilitating the development of more sophisticated and robust driving ecosystems [5, 25]. However, addressing challenges such as data transmission latency and bandwidth limitations is crucial to fully realizing the potential of this integration [26]. This underscores the importance of ongoing research and investment efforts to maximize the advantages of combining cloud technologies with a connected and automated driving environment [1, 27, 28]. Testing and validating such an integrated system with real-time applications requires a comprehensive evaluation to ensure compliance with latency constraints and other application requirements [16, 29]. However, many difficulties arise from the need for vehicles to be equipped with sensors and communication hardware and software modules that integrate with cloud servers [4, 26, 30]. This not only incurs high costs, but also faces limitations due to safety and government regulations. Consequently, the evaluation framework and the design of the real world tests are constrained, leading researchers to rely on simulation-based studies [31] lacking real-world use cases [27].

III. Experimental Setup

In this section, we detail the methodology used to design and setup the testbed framework aimed at evaluating distributed vehicular perception.

A. *Tested Scenarios*

We evaluate the performance of cooperative object detection through two distinct scenarios, depicted on Fig. 1.

- **Local scenario:** Camera images are processed on-board upon capture. To cope with on-board hardware constraint, a lightweight detector is used (see Section III-F for details). The detection results are then encoded to generate CPMs, which are broadcast to nearby vehicles or roadside units (RSU) using the ITS-5G communication standardized by the The European Telecommunications Standards Institute (ETSI) [32].
- **Cloud scenario:** Camera images are compressed on-board upon capture using the H.265 codec. The compressed bitstream is then transmitted to the cloud using the cellular mode of C-V2X. Once on the cloud, the encoded frames undergo decoding, followed by an inference step to detect objects. Since hardware constraints are relaxed on the cloud compared to on-board, a more computationally demanding model is used than in the Local scenario. Finally, the detection results are transmitted back to the originating vehicle as well as to nearby vehicles using C-V2X.

The Local scenario focuses on on-board processing within the vehicle, while the Cloud scenario explores the potential benefits of cloud-based processing. We aim to evaluate and compare the performance of local and cloud processing approaches in terms of end-to-end delay, detection quality, and overall effectiveness in real-world driving scenarios.

B. *Vehicle Instrumentation*

We conduct experiments using an experimental vehicle used for connected and autonomous driving research. The vehicle, a 2018 KIA Soul EV, is equipped with a suite of sensors, including a LiDAR, cameras, and a GNSS receiver. These sensors are crucial for capturing environmental data required for perception tasks. However, for the purposes of this study, we focus exclusively on processing data from a single front-facing camera sensor. Communications between vehicles and with infrastructure are facilitated by the YoGoKo Y-Box communication module, which features ITS-G5 and C-V2X

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.

communication technologies. For further information on the test vehicle, equipped sensors, hardware, and driving software stack, we refer the reader to [33].

C. Test Route & Network Measurements

Our driving tests were conducted on public roads in the Kirchberg area of Luxembourg City. It includes diverse road features and traffic conditions, allowing us to evaluate different cooperative perception scenarios at the highest level of realism.

In the Local scenario, the experimental setup for the CPM measurements involved conducting V2X communication using the ITS-G5 protocol within a dynamic environment. The experimental setup involved a stationary receiver located at specific coordinates (lon 6.161993, lat 49.626478) while a transmitter vehicle moved at speeds ranging from 40 to 50 km/h. This stationary receiver position was chosen to ensure consistency in measurement conditions, providing a fixed reference point for assessing the quality of communication between the transmitter vehicle and the receiver. The experimental route spans over a distance of approximately 1.5km[2]. Within this experimental setup, we evaluated the transmission of CPM messages to assess the reliability and performance of V2X communication under public traffic conditions.

In the Cloud scenario, the driving route spans approximately 4km[3] and we use the cellular mode of the C-V2X technology to communicate with a cloud server located at the premises of the University of Luxembourg in the same area. Twelve commercial base stations are located in proximity of the test route, including 4G and 5G (non-standalone) cellular sites, using Low- (700 MHz) and Mid-band (3.6 GHz) frequencies. The area features an average download throughput of 55 Mbps for 4G and 105 Mbps for 5G, whereas the average upload throughput oscillates between 20 and 30 Mbps for both technologies. We use UDP for data offloading to the cloud, as UDP has shown potential for streaming sensor data to a remote cloud for processing with the lowest end-to-end delay [34].

D. Data Collection & Annotation

Data collection is a crucial aspect in the development and evaluation of perception systems for autonomous vehicles. To obtain more accurate detection results on the previously introduced test routes, we first collect and annotate a dataset, and then perform transfer learning on the pre-trained YOLO model. To do so, we set up a front-facing camera and a GNSS sensor mounted in the front of the roof rack of the vehicle. We collected 5000 frames at 1Hz with synchronized time and GNSS measurements along the Cloud test route described in Section III-C. To generate ground truth data for training and evaluation purposes, we manually annotated the collected frames and divided them into training (70%), validation (15%), and test (15%) subsets. The annotation process involved meticulously labeling objects of interest in the captured images, focusing on three object classes: pedestrians, vehicles, and traffic lights. We use the specialized Roboflow [35] annotation tool to streamline this process and ensure accurate labeling. The dataset comprises a total of 59574 annotated instances across all classes, with an average of 15 objects per image.

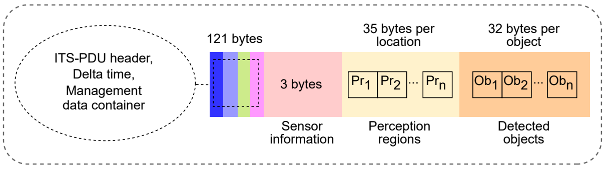


Fig. 2: A basic overview of different containers included in the CPM message format as defined by the ETSI standard [32].

E. CPM Encoding

In the Local scenario, the results of the object detection performed on-board are packaged into CPMs and broadcast to nearby vehicles over the ITS-G5 communication protocol. The list of detected objects is encoded into CPMs following the ETSI standard [32]. A basic overview of different containers included in the CPM message format is displayed on Fig. 2. The ITS-PDU header, management data, and sensor information containers include, among others, information such as protocol versions, message type, reference position, and sensor ID. Containers of detected objects include information about the detected objects, such as the class of the object and the confidence level assigned by the perception system. According to the ETSI definition, the CPM should also include perception regions, which shall list the location information of the detected objects calibrated with the sensor location. Obtaining the location of the object would require an additional localization technique to be employed for each detected object, which is beyond the scope of this work. In order to maintain the standard format of the CPM payload, the ego vehicle location is used for the perception region.

F. Hardware Configuration & Detection Models

The limited processing capabilities and energy constraints inherent to vehicles pose significant challenges to real time applications such as cooperative perception. As a result, computational tasks are often offloaded to external resources, where computing resources are more abundant. However, using this paradigm, achieving high detection quality and low end-to-end delay requires a well-designed architecture, high speed networks and high-performance computing [36]. The hardware configurations used in our experiments are detailed in Table I. The Local setup refers to the on-board device, for which we select hardware with appropriately limited power consumption. In contrast, Cloud offers more robust compute capabilities, at higher financial and electrical power costs.

To perform object detection, we use YOLOv8 [37], for its state-of-the-art performance in terms of accuracy and speed [10]. YOLOv8 offers various models ranging from small to extra-large. Table II presents a comparison of YOLOv8 variants in terms of model complexity, inference speed (Local and Cloud), and detection accuracy. The comparison highlights

[2] **Local test route:**http://g-o.lu/3/GsHC
[3] **Cloud test route:**http://g-o.lu/3/96TS

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.

Platform	Hardware configuration
Local (≈100W)	GeForce GTX 1650 GPU 896 CUDA Cores, 5.7 TFLOPS (FP16) Intel i9-9980HK @2.4GHz
Cloud (≈450W)	NVIDIA GeForce RTX 4090 GPU 16384 CUDA Cores, 82.6 TFLOPS (FP16) Intel i9-13900K @2.8GHz

TABLE I: Hardware setups for the Local and Cloud scenarios described in Section III-A.

significant differences in inference times between Local and Cloud processing. While Local inference times range from 19.2 to 521.4 ms depending on the model variant, cloud-based inference is consistently faster, ranging from 1.2 to 3.9 ms. These results underscore the advantage of offloading computational tasks to a cloud where power constraints are lifted and resources are abundant, allowing the use of better models while significantly reducing inference time. Since our study focuses on real-time detection, we select the *small* variant as the lightweight detector running on-board in our Local scenario. For the Cloud scenario where power constraints are relaxed, we use the much more demanding *xlarge* variant.

YOLOv8 variant	Params(M)	FLOPS(B)	Inference(ms)		mAP
			Local	Cloud	
nano	3.2	8.7	19.2	1.2	0.65
small	11.2	28.6	**29.4**	1.9	0.69
medium	25.9	78.9	192.3	2.7	0.74
large	43.7	165.2	361.1	3.2	0.85
xlarge	68.3	257.8	521.4	**3.9**	0.89

TABLE II: Comparison of YOLOv8 variants in terms of model complexity, inference time, and mean average precision (mAP) on the test split of the dataset presented in Section III-D.

G. Model Training

We trained the *small* and *xlarge* YOLOv8 variants, for the Local and Cloud scenarios respectively. We split the dataset described in Section III-D into a training set 3500 frames, a validation set of 750 frames, and a test set 750 frames. To avoid class imbalance, we include the same proportion of each class into each dataset split. All models are trained for up to a 100 epochs with early stopping and using the Adam optimizer with an initial learning rate of 0.001. Rather than learning from scratch, we leverage transfer learning by initializing models with pre-trained weights obtained by training on the COCO dataset [38]. To accelerate training, we set the batch size to the maximal value that fits in GPU memory.

IV. Results and Discussion

In this section, we present the results of our experiments and provide a detailed discussion of the findings. The experiments were conducted under typical day weather conditions featuring partly cloudy skies, on a predefined route as detailed in Section III-C. To ensure the collection of statistically valid results, each experiment consisted of 10 repetitions.

A. ITS-G5 CPM Transmission (Local scenario only)

The CPM transmission latency is influenced by various factors, such as network conditions, signal quality, and hardware capabilities, including processing time at the receiving end. These factors require a detailed investigation to understand their collective impact on the performance of CPM transmission. To ensure reliable transmission, we begin by measuring RSSI while transmitting CPM over ITS-G5. Our findings, as shown in Fig. 3a, reveal a consistent pattern in which RSSI values exhibit an inverse correlation with distance. As expected, with increasing distance between the sender and receiver, the RSSI decreases. However, in particular, beyond a distance of 150 m, we observed an increase in path loss. This phenomenon suggests additional factors influencing the dynamics of signal propagation beyond a distance of 150 m. Environmental variables such as buildings, terrain topology and signal interference may contribute to this observed increase in path loss [19]. Despite this deviation, the results of the experiments demonstrate reliable transmission range of 150 m, and a maximum range of up to 400 m. The network parameters with transmission power and radio configuration details are summarized in Table III.

Parameter Name	Value
Transmission Power (Tx)	23 dBm
Energy threshold	-85 dBm
Channel bandwidth / carrier frequency	10 MHz / 5.9 GHz
Radio Configuration	Single Channel (CCH)
Data rate	6 Mbps
Number of CPM Transmitted / loss ratio	4000 / 0.07

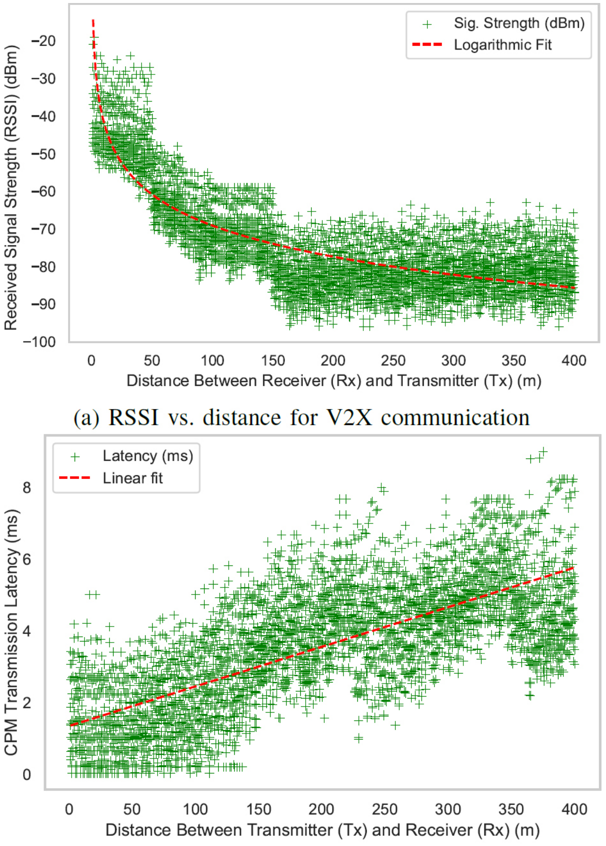
TABLE III: ITS-G5 network parameters.

In a second experiment, we measured end-to-end CPM transmission delay between the mobile sender and static receiver. The results are shown in Fig. 3b. It shows that the CPM transmission latency remains consistently low, with values ranging from 1.5 (± 1.0) ms to 6 ms (±2.5) ms, for a communication range of up to 400 m. These findings align with previous studies on ITS-G5 technologies, which have demonstrated low latency and reliable transmission of CPMs in a simulated connected driving scenarios [39].

B. Video Streaming Latency (Cloud scenario only)

To stream camera images for cloud detection, we consider two different streaming resolutions: 1920x1080 (FHD) and 1280x720 (HD). Encoding always occurs in the on-board hardware, while decoding is performed on the cloud. Depending on the compression quality, the latency for encoding and decoding ranges from 0.5 to 3 ms, which is relatively low compared to inference and transmission times. The choice of compression value of H.265, controlled by Constant Rate Factor (CRF) while keeping other parameters to defaults [40], results in streaming latencies ranging from 3 to 457.4 ms for FHD and 2.5 to 341.4 ms for HD, as illustrated in Fig. 4. The CRF values range from 0 to 51, where 0 corresponds to lossless compression and 51 represents the highest possible compression with the greatest loss of data.

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.



(a) RSSI vs. distance for V2X communication

(b) Latency vs. distance for V2X communication

Fig. 3: Demonstration of (a) RSSI measurements and (b) latency measurements, plotted against distance between receiver and transmitter during CPMs transmission using the ITS-G5.

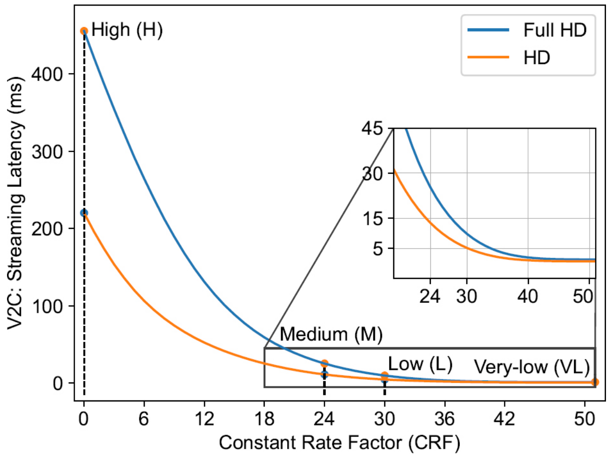


Fig. 4: For cloud processing, we measured the mean latency for HD and FHD input resolutions at various H.265 compression qualities: High (H), Medium (M), Low (L) and Very Low (VL). The results focus solely on vehicle-to-cloud transmission using UDP, excluding encoding and decoding latencies.

To further investigate the end-to-end delay of the cloud scenario, we define distinct settings: FHD-H (High quality) at CRF 0, FHD-M (Medium quality) at CRF 24, FHD-L (Low quality) at CRF 30, and FHD-VL (Very Low quality) at CRF 51. Similarly, we assign the same CRF values for HD, defining the following settings: HD-H, HD-M, HD-L and HD-VL.

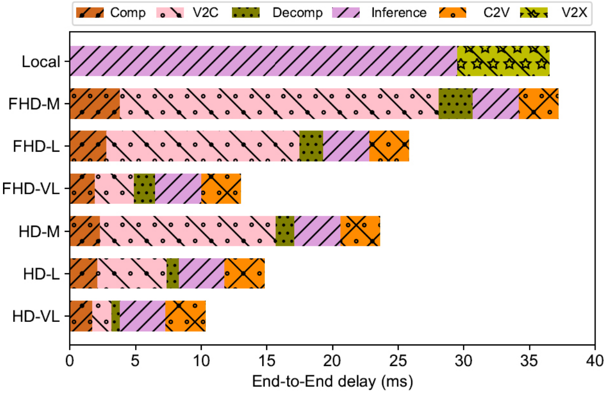


Fig. 5: Breakdown of mean end-to-end delay (ms) for Local and Cloud processing scenarios. In the Local scenario, end-to-end delay depends solely on inference time and V2X (CPM) communication. For the Cloud processing, end-to-end delay depends on compression, vehicle-to-cloud (V2C) streaming, decompression, inference in the cloud, and the Cloud-to-vehicle (C2V) communication latency.

C. *End-to-End Delay*

In this section, we examine the end-to-end delay of both, the Local and the Cloud scenario. A breakdown of the different components of end-to-end delays for the different experiments is displayed on Fig. 5. In the case of the Local scenario, the end-to-end delay depends on only the on-board inference time of 29.4 ms (see Table II) and on the V2X latency incurrent by CPM transmission over ITS-G5 (see Fig. 3b). Hence, the local end-to-end delay reaches 36.4 ms.

In the Cloud processing scenario, the end-to-end delay is dependent on a series of interconnected factors. These include compressing images using H.265, latency incurred during vehicle-to-cloud (V2C) communication over C-V2X in cellular mode, decoding executed within the cloud infrastructure, inference time for object detection, and transmission of detection results back to one or more vehicles (C2V) in the vicinity.

Conversely, in the Cloud processing scenario, considering FHD-M, the delay can increase to 40.8 ms, while considering HD-M, the delay decreases to 30.4 ms. For FHD-VL and HD-VL streaming, the delay decreases to 20.8 and 15.26 ms, respectively. In terms of end-to-end delay, all compression factors are viable for real-time operation, except for high compression qualities (FHD-H and HD-H). In the cases of FHD-H and HD-H, not displayed on Fig. 5 for scaling reasons, delay respectively reaches 457.4 and 341.4 ms. Note that inference time remains constant at 3.9 ms for all Cloud experiments since the same detection model is used.

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.

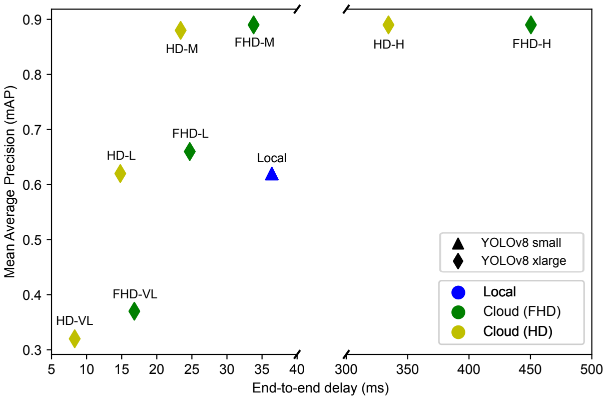


Fig. 6: Comparison of mAP against End-to-end Delay for local and cloud-based processing, for two distinct input resolutions (HD and FHD), and four compression qualities ranging from High (H) to Very Low (VL).

D. Detection Quality vs. End-to-End Delay Trade-off

To gain a comprehensive understanding of the proposed distributed and cooperative perception strategies, it is essential to carefully consider the trade-off between detection quality and end-to-end delay. By exploring this trade-off, our objective is to identify the optimal balance between quality and end-to-end delay. Fig. 6 synthesizes our results by presenting different trade-offs between end-to-end delay and mean average precision (mAP), for Local and Cloud processing with different input resolutions and compression qualities. In Cloud processing scenarios, a higher mAP score of 0.89 is achieved in FHD-H with an end-to-end delay of 450.4 ms. For high-quality compression scenarios (FHD-H and HD-H), the use of CRF 0 entails lossless compression and results in larger amount of data streamed to the cloud. This inevitably increases end-to-end delay in both the FHD-H (450.4ms) and HD-H (334.5ms) cases. With optimal choices of compression qualities, end-to-end delays can be lowered below 50 ms without disproportionately sacrificing mAP. Specifically, the FHD-M and HD-M scenarios maintain a high accuracy compared to their high compression quality counterparts (0.89 mAP from FHD-H to FHD-M, 0.88 mAP from HD-H to HD-M), while drastically reducing end-to-end delay (33.8 ms from FHD-H to FHD-M, 23.4 ms from HD-H to HD-M). A further reduction is possible by streaming Very Low quality images (HD-VL and FHD-VL), but this results in degraded mAP compared to even the lightweight Local model. Since the HD-L model provides substantially higher mAP at a comparable end-to-end delay, there is no tangible benefit of using Very Low compression.

The visual impact of lowering compression quality and its effect on detection results are illustrated on Fig. 7. In particular, in FHD-M settings, Cloud detection rates closely resemble those of object detection, in contrast to FHD-H settings. However, a decrease in detection instances begins with the FHD-L scenario. In FHD-VL, only a traffic light is detected, leaving other objects undetected, thereby compromising overall detection quality due to the low-quality stream.

V. Conclusion and Future Work

In this study, we analyze real-time object detection strategies, which include both Local and Cloud-based methodologies, and employ field trials for validation. We demonstrate the feasibility of ITS-G5 for robust transmission of CPM in local environments. Furthermore, we explore the potential of C-V2X communication to utilize cloud hardware, facilitating real-time object detection. Our findings indicate significantly improved detection quality compared to processing solely onboard hardware. To gain a comprehensive understanding of the proposed cooperative perception strategies, we investigate the trade-off between detection quality and end-to-end delay. In addition, we also create and release a real-world driving dataset annotating instances of pedestrian, vehicle, and traffic light classes providing a valuable resource for researchers. Future research will explore adaptive techniques and hybrid optimization strategies, including dynamically switching between local and cloud processing based on bandwidth availability and situational complexity, such as high-accuracy demands in areas like intersections.

Acknowledgments

This work is supported by the Fonds National de la Recherche of Luxembourg (FNR), under AFR grant agreement No 17020780 and project acronym *ACDC*.

References

[1] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 252–1726.

[2] A. Yaqoob, T. Bi, and G.-M. Muntean, "A survey on adaptive 360 video streaming: Solutions, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2801–2838, 2020.

[3] C. Liu, Y. Chen, J. Chen, R. Payton, M. Riley, and S.-H. Yang, "Cooperative perception with learning-based v2v communications," *IEEE Wireless Communications Letters*, vol. 12, no. 11, pp. 1831–1835, 2023.

[4] B. McCarthy and A. O'Driscoll, "Opencv2x mode 4: A simulation extension for cellular vehicular communication networks," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–6.

[5] S. Gyawali, S. Xu, Y. Qian, and R. Q. Hu, "Challenges and solutions for cellular based v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 222–255, 2020.

[6] F. Hawlader, F. Robinet, and R. Frank, "Leveraging the edge and cloud for v2x-based real-time object detection in autonomous driving," in *Computer Communications*, vol. 213, 2024, pp. 372–381.

[7] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, 1992.

[8] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[9] F. Hawlader, F. Robinet, and R. Frank, "Vehicle-to-infrastructure communication for real-time object detection in autonomous driving," in *2023 18th Wireless On-Demand Network Systems and Services Conference (WONS)*.

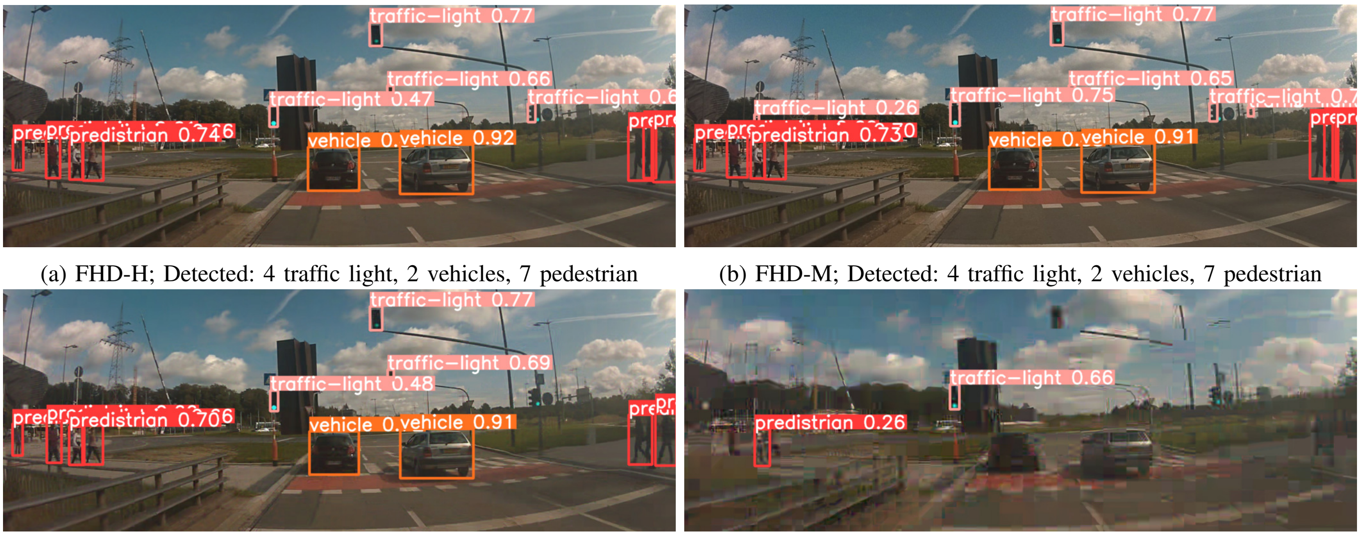
[10] I. Lazarevich, S. Grimaldi, and S. Sah, "Yolobench: Benchmarking efficient object detectors on embedded systems," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[11] M. Řeřábek and T. Ebrahimi, "Comparison of compression efficiency between hevc/h. 265 and vp9 based on subjective assessments," in *Applications of digital image processing XXXVII*. SPIE, 2014.

[12] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Cooperative perception for connected and automated vehicles: Evaluation and impact of congestion control," *IEEE Access*, vol. 8, pp. 197 665–197 683, 2020.

[13] Y. Asabe, E. Javanmardi, J. Nakazato, M. Tsukada, and H. Esaki, "Autowarev2x: Reliable v2x communication and collective perception

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.



(a) FHD-H; Detected: 4 traffic light, 2 vehicles, 7 pedestrian

(b) FHD-M; Detected: 4 traffic light, 2 vehicles, 7 pedestrian

(c) FHD-L; Detected: 2 traffic light, 2 vehicles, 6 pedestrian

(d) FHD-VL; Detected: 1 traffic light, 0 vehicles, 0 pedestrian

Fig. 7: Cloud-based detection of pedestrians, vehicles, and traffic lights in FHD at various compression qualities. Detections are considered valid when they have 50%+ IoU with the ground truth. **Ground truth:** 4 traffic lights, 2 vehicles and 7 pedestrians.

for autonomous driving," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–7.

[14] F. Hawlader and R. Frank, "Towards a framework to evaluate cooperative perception for connected vehicles," in *2021 IEEE Vehicular Networking Conference (VNC)*, 2021, pp. 36–39.

[15] S.-W. Kim, B. Qin, Z. J. Chong, X. Shen, W. Liu, M. H. Ang, E. Frazzoli, and D. Rus, "Multivehicle cooperative driving using cooperative perception: Design and experimental validation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 663–680, 2015.

[16] V. Maglogiannis, D. Naudts, S. Hadiwardoyo, D. van den Akker, J. Marquez-Barja, and I. Moerman, "Experimental v2x evaluation for c-v2x and its-g5 technologies in a real-life highway environment," *IEEE Transactions on Network and Service Management*, vol. 19, no. 2, 2022.

[17] J. Stellwagen, M. Deegener, and M. Kuhn, "Application-dependent support of its-g5 car-to-car communication by selected usage of lte-v2x resources," in *Automotive meets Electronics*, 2021.

[18] P. Große, C. Andrich, W. Kotterman, A. Ihlow, and G. D. Galdo, "Measuring etsi its-g5 communications latencies with commercial off-the-shelf wi-fi hardware," in *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, 2018, pp. 1–4.

[19] K. Garlichs, H.-J. Günther, and L. C. Wolf, "Generation rules for the collective perception service," in *2019 IEEE Vehicular Networking Conference (VNC)*, 2019, pp. 1–8.

[20] T. Dias, E. Silva, J. Almeida, and J. Ferreira, "An its-g5 v2x solution in c-roads portugal," *Transportation Research Procedia*, 2023.

[21] H. Zhou, W. Xu, J. Chen, and W. Wang, "Evolutionary v2x technologies toward the internet of vehicles: Challenges and opportunities," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 308–323, 2020.

[22] K. A. Khaliq, O. Chughtai, A. Shahwani, A. Qayyum, and J. Pannek, "Road accidents detection, data collection and data analysis using v2x communication and edge/cloud computing," *Electronics*, 2019.

[23] W. Feng, S. Lin, N. Zhang, G. Wang, B. Ai, and L. Cai, "C-v2x based offloading strategy in multi-tier vehicular edge computing system," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*.

[24] L. Liu, S. Lu, R. Zhong, B. Wu, Y. Yao, Q. Zhang, and W. Shi, "Computing systems for autonomous driving: State of the art and challenges," *IEEE Internet of Things Journal*, 2020.

[25] B. Flowers, Y.-J. Ku, S. Baidya, and S. Dey, "Utilizing reinforcement learning for adaptive sensor data sharing over c-v2x communications," *IEEE Transactions on Vehicular Technology*, 2023.

[26] M. Ahmed, H. Mirza, F. Xu, W. U. Khan, Q. Lin, and Z. Han, "Vehicular communication network enabled cav data offloading: A review," *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[27] J. E. Naranjo, F. Jimenez, J. J. Anaya, R. Castiñeira, M. Gil, and D. Romero, "Validation experiences on autonomous and connected driving in autocits pilot in madrid," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, 2018, pp. 1–4.

[28] F. Hawlader and R. Frank, "Realistic cooperative perception for connected and automated vehicles: A simulation review," in *2023 8th International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2023, pp. 1–6.

[29] G. Yu, H. Li, Y. Wang, P. Chen, and B. Zhou, "A review on cooperative perception and control supported infrastructure-vehicle system," *Green Energy and Intelligent Transportation*, p. 100023, 2022.

[30] S. A. Yusuf, A. Khan, and R. Souissi, "Vehicle-to-everything (v2x) in the autonomous vehicles domain–a technical review of communication, sensor, and ai technologies for road user safety," *Transportation Research Interdisciplinary Perspectives*, vol. 23, p. 100980, 2024.

[31] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, January 2011.

[32] E. E. TS, "103 324 v2. 1.1; intelligent transport system (its); vehicular communications; basic set of applications; collective perception service," *ETSI: Sophia Antipolis, France*, 2023.

[33] G. Varisteas and R. Frank, "Poster: Junior, a research platform for connected and automated driving," in *2019 IEEE Vehicular Networking Conference (VNC)*, 2019, pp. 1–2.

[34] A. B. De Souza, Rego., P. P. R. Filho, J. N. De Souza, V. Chamola, V. H. C. De Albuquerque, and B. Sikdar, "Computation offloading for vehicular environments: A survey," *IEEE Access*, 2020.

[35] G. Jocher, A. Stoken *et al.*, "ultralytics/yolov5n'nano'models, roboflow integration, tensorflow export, opencv dnn support," *Zenodo*, 2021.

[36] G. Zhu, Z. Lyu, X. Jiao, P. Liu, M. Chen, J. Xu, S. Cui, and P. Zhang, "Pushing ai to wireless network edge: An overview on integrated sensing, communication, and computation towards 6g," *Science China Information Sciences*, vol. 66, no. 3, p. 130301, 2023.

[37] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[38] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll'a r, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[39] M. Karoui, A. Freitas, and G. Chalhoub, "Performance comparison between lte-v2x and its-g5 under realistic urban scenarios," in *2020 IEEE 91st vehicular technology conference (VTC2020-Spring)*. IEEE.

[40] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, 2012.

Authorized licensed use limited to: University of Luxembourg. Downloaded on December 17,2024 at 23:04:36 UTC from IEEE Xplore. Restrictions apply.